

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-61. (Canceled)

62. (Currently Amended) A method by which a disk-based distributed data storage system is organized for protecting historical records of stored data entities, the method comprising:

recording distinct states of stored data entities, corresponding to different moments of time, as a plurality of entity versions coexisting within the distributed data storage system;

storing copies of an entity version that is one of the plurality of entity versions at each of a plurality of storage sites of the distributed data storage system;

sharing, among the plurality of storage sites, a set of rules that restrict deletion of the entity version; and

applying the shared set of rules independently at each of the plurality of storage sites, to determine whether or not the copies of the entity version can be deleted;

wherein if it is determined that the copies of the entity version cannot be deleted then they also cannot be modified;

~~wherein deletion of the copies of the entity version from the distributed data storage system is only allowed if the set of rules determine that the entity version can be deleted; and~~

wherein a client program communicating with the disk-based distributed data storage system deposits the entity version into the storage system;

wherein [an] a first action-taken request sent by [a] the client program communicating with the disk-based distributed data storage system causes the shared set of rules to restrict deletion of the entity version;

wherein a second request, sent by the client program after the first request, would enable the entity version to be deleted in violation of the restriction, and the request is denied;

wherein [and] no subsequent action-taken request sent by the client program can cause enable deletion to occur in violation of the restriction;

wherein the sharing step comprises storing at each of the plurality of storage sites information derived from the set of rules;

wherein the independent application of the shared set of rules at each of the plurality of storage sites is designed to prevent alterations or corruptions of the operation at a one of the plurality of storage sites from allowing the entity version to be deleted or modified at another of the plurality of storage sites in violation of the restriction on the deletion of the entity version;

wherein the plurality of storage sites communicate with one another in order to achieve fault tolerance against the loss of storage sites;

wherein the shared set of rules restrict deletion, based at least in part upon a time that was associated with the entity version by an act of the client program; and

wherein a third request, sent by the client program after the second request, causes the copies of the entity version to be deleted from the plurality of storage sites.

63. (Canceled)

64. (Currently Amended) The method of claim 62 in which the time is an expiration time[s are] assigned to the entity version, and the assignment is made independently within each of the plurality of storage sites, according to the shared set of rules, before which times both modification and deletion are prohibited.

65. (Original) The method of claim 62 in which no single individual is given the authority to override the deletion prohibition at all of the plurality of storage sites.

66-163. (Canceled)

164. (Currently Amended) The method of claim 62 in which applying the set of rules at a one of the plurality of storage sites determines that [the] an entity version can be deleted and a copy of the entity version is deleted immediately and storage space that was used to store the copy becomes available to store new data.

165. (Currently Amended) The method of claim 62 in which applying the set of rules determines that ~~the~~ an entity version can be deleted but ~~the copies of the entity version are only is~~ not deleted if until deletion is requested by a client of the disk-based distributed data storage system.

166. (Canceled)

167. (Previously Presented) The method of claim 62 in which, during a time interval, the shared set of rules prohibits deletion of the entity version while others of the plurality of entity versions are allowed to be deleted.

168. (Currently Amended) The method of claim 167 in which the time interval is at least a year in length.

169. (Canceled)

170. (Previously Presented) The method of claim 167 in which the client program causes the time interval during which deletion is prohibited to be extended and no subsequent action taken by the client program can cause the time interval to be shortened.

171. (Previously Presented) The method of claim 167 in which the client program causes the length of the time interval to be set and no subsequent action taken by the client program can shorten the time interval.

172. (Previously Presented) The method of claim 171 in which the length of the time interval is initially not set and, before the length of the time interval is set, no action taken by the client program can cause the entity version to be deleted

173. (Previously Presented) The method of claim 167 in which no action taken by any client program that only communicates with the disk-based distributed data storage system over a wide area network can cause the time interval to be shortened.

174. (Previously Presented) The method of claim 62 in which the plurality of entity versions record historical states of a single stored data entity, with each of the plurality of entity versions associated with a historical time interval during which the recorded historical state was the state of the single stored data entity.

175. (Previously Presented) The method of claim 174 in which the shared set of rules that determine whether or not the entity version can be deleted depend at least in part on the length of the historical time interval associated with the entity version.

176. (Previously Presented) The method of claim 174 in which the shared set of rules that determine whether or not the entity version can be deleted depend at least in part on whether or not the historical time interval associated with the entity version includes a specified moment of time.

177. (Previously Presented) The method of claim 62 in which a stored data entity is a file in a file system or a record in a database or an object in an object storage system.

178. (Currently Amended) The method of claim 62 in which two of the plurality of storage sites are ~~at least a mile apart~~ located in different cities.

179. (Currently Amended) The method of claim 62 in which the set of rules comprise a ~~program~~ rule description that is separate and distinct from the software that implements the disk-based distributed data storage system, and sharing occurs at the time when the plurality of entity versions are being stored in the storage system, and the information derived from the set of rules that is stored at each of the plurality of storage sites comprises a hash of the description.

180. (New) The method of claim 62 in which the entity version is a version of a stored data entity and the first request causes a new version of the stored data entity to be stored.

181. (New) The method of claim 62 in which the first request assigns an expiration time to the entity version, before which time deletion is prohibited.

182. (New) The method of claim 62 in which the second or third request attempts to delete the entity version or to change the time associated with the entity version.

183. (New) The method of claim 62 in which the entity version is a version of a stored data entity and the third request causes a new version of the stored data entity to be stored.

184. (New) The method of claim 62 in which the time associated with the entity version is a time when the entity version was created, transmitted or stored; or had some property changed; or a time assigned to the entity version.

185. (New) The method of claim 62 in which the shared set of rules are communicated to the plurality of storage sites at the time that the client program communicating with the disk-based distributed data storage system deposits the entity version into the storage system.